

Maximizing Coverage Centrality via Network Design: Extended Version

Sourav Medya¹, Arlei Silva¹, Ambuj Singh¹, Prithwish Basu², Ananthram Swami³

¹Computer Science Department, University of California, Santa Barbara, CA, USA

²Raytheon BBN Technologies, Cambridge, MA, USA, ³Army Research Laboratory, Adelphi, MD, USA

medya,arlei,ambuj@cs.ucsb.edu, pbasu@bbn.com, ananthram.swami.civ@mail.mil

ABSTRACT

Network centrality plays an important role in many applications. Central nodes in social networks can be influential, driving opinions and spreading news or rumors. In hyperlinked environments, such as the Web, where users navigate via clicks, central content receives high traffic, becoming targets for advertising campaigns. While there is an extensive amount of work on centrality measures and their efficient computation, controlling nodes' centrality via network updates is a more recent and challenging problem. Performing minimal modifications to a network to achieve a desired property falls under the umbrella of network design problems. This paper is focused on improving the coverage centrality of a set of nodes, which is the number of pairs of nodes that have a shortest path passing through the set, by adding edges to the network. We prove strong inapproximability results and propose a greedy algorithm for maximizing coverage centrality. To ensure scalability to large networks, we also design an efficient sampling algorithm for the problem. In addition to providing an extensive empirical evaluation of our algorithms, we also show that, under some realistic constraints, the proposed solutions achieve almost-optimal approximation for coverage centrality maximization.

KEYWORDS

Coverage Centrality, Network Design, Approximation Algorithms

ACM Reference format:

Sourav Medya¹, Arlei Silva¹, Ambuj Singh¹, Prithwish Basu², Ananthram Swami³. 2017. Maximizing Coverage Centrality via Network Design: Extended Version. In *Proceedings of ACM KDD conference, Halifax, Nova Scotia - Canada, August 2017 (KDD'17)*, 10 pages. DOI: 10.475/123.4

1 INTRODUCTION

Network design is a recent area of study focused on modifying or redesigning a network in order to achieve a desired property [10, 32]. As networks become a popular framework for modeling complex systems (e.g. VLSI, transportation, communication, society), network design provides key controlling capabilities over these systems, specially when resources are constrained. Existing work has investigated the optimization of global properties, such as minimum spanning tree [16], shortest-path distances [8, 18, 21],

diameter [7], and information diffusion-related metrics [14, 29] via a few local (e.g. vertex, edge-level) upgrades in the network. Due to the rapid growth of data, computing a global property of a network becomes time-intensive. For instance, computing all-pair shortest paths in large networks is prohibitive. As a consequence, design problems in large networks are inherently challenging. Moreover, because of the combinatorial nature of these local modifications, network design problems are often NP-hard, and thus, require the development of efficient approximation algorithms.

We focus on a novel network design problem, which is improving the *coverage centrality* of a group of nodes. Given a node v , its coverage centrality is the number of distinct node pairs for which a shortest path passes through v . The centrality of a group X measures the total number of node pairs for which shortest paths go through members of X . *Our goal is to maximize group centrality, for a given target group of nodes, via a small number of edge additions.*

As an application scenario, consider an online advertising service where advertisers can place links on a set of pages depending on user context information (see Figure 1). For instance, users navigating from travel to car related pages are likely to be interested in car rentals. Thus, the ad service can display links in a subset of pages in order to increase the number of shortest paths from travel related web-pages to car related ones via a set of pages owned by a given car rental company. The idea is to boost the traffic to the car rental pages while users browse the Web, assuming that clicks will often follow shortest paths. Once the user arrives at an advertiser's page, the car rental company can offer targeted information to support her browsing through the automobile related content (e.g. highlighting car models that are often rented in a given tourist location). This problem is equivalent to optimizing the coverage centrality of the advertiser's pages—for a selected set of node pairs—by adding few edges from a candidate set to the Web graph.

Another application scenario is a professional network, such as *LinkedIn*, where the centrality of some users (e.g. employees of a given company) might be increased via connection recommendations/advertising. In military settings, where networks might include adversarial elements, inducing the flow of information towards key agents can enhance communication and decision making [26]. Moreover, multiple recent approaches focus on the most probable (shortest) paths to allow scalable solutions [4, 15] for social influence and information propagation. Thus, to achieve better information propagation through a target group of nodes of interest, one needs to improve their shortest path based centrality.

From a theoretical standpoint, for any objective function of interest, we can define a *search* and a corresponding *design* problem. In this paper, we show that coverage centrality maximization is NP-hard, even to approximate by a constant, in contrast with the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

KDD'17, Halifax, Nova Scotia - Canada

© 2016 Copyright held by the owner/author(s). 123-4567-24-567/08/06...\$15.00

DOI: 10.475/123.4

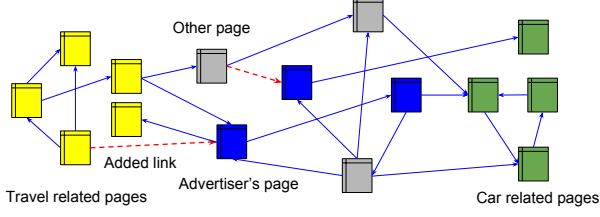


Figure 1: Application scenario: Adding links to the Web graph in order to increase user traffic from travel to car related pages via a set of car rental (advertiser's) pages.

search version of the problem, which is also NP-hard but can be approximated by a simple greedy algorithm [31]. This result shows that, for coverage centrality, the design problem has a stronger combinatorial nature than its search counterpart. Furthermore, we define two realistic constraints for our general problem and show that, under these constraints, the problem is APX-hard but allows a constant factor approximation by a greedy algorithm. In fact, we are able to show that our approximation for the constrained problem is *almost optimal*, in the sense that the best algorithm cannot achieve an approximation significantly far from the one obtained by our approach. In order to scale our greedy approach to large datasets, we also propose an efficient sampling scheme, with approximation guarantees, for coverage centrality maximization.

Previous Work. There is a considerable amount of research on network design targeting various objectives by modifying the network structure and node/edge attributes. These problems differ mostly on the upgrading models and objective functions considered.

General network design problems: A set of design problems were introduced by Paik et al. [23]. They focused on vertex upgrades to improve the delays on adjacent edges. Krumke et al. [16] generalized this model and proposed minimizing the cost of the minimum spanning tree with varying upgrade costs for vertices/edges. Lin et al. [18] also proposed a shortest path optimization problem via improving edge weights under a budget constraint and with undirected edges. In [8, 20], the authors studied a different version of the problem, where weights are set to the nodes.

Design problems via edge addition: Meyerson et al. [21] proposed approximation algorithms for single-source and all-pair shortest paths minimization. Faster algorithms for the same problems were presented in [24]. Demaine et al. [7] minimized the diameter of a network and node eccentricity by adding shortcut edges with a constant factor approximation algorithm. Past research had also considered eccentricity minimization in a composite network [26]. However, all aforementioned problems are based on improving distances and hence are complementary to our objective.

Centrality computation and related optimization problems: This line of research is the most related to the present work. The first efficient algorithm for betweenness centrality computation was proposed by Brandes [2]. Recently, [27] introduced an approach for computing the top- k nodes in terms of betweenness centrality via VC-dimension theory. Yoshida [31] studied similar problems—for both betweenness and coverage centrality—in the adaptive setting, where shortest paths already covered by selected nodes are not taken into account. Yoshida's algorithm was later improved using a different sampling scheme [19]. Here, we focus on the design

Symbols	Definitions and Descriptions
$d(s, t)$	Shortest path (s.p.) distance between s and t
n	Number of nodes in the graph
m	Number of edges in the graph
$G(V, E)$	Given graph (vertex set V and edge set E)
X	Target set of nodes
$C(v), C(X)$	Coverage centrality of node v , node set X
Γ	Candidate set of edges
k	budget
P_{st}	The set of nodes on the s.p.s between s and t
G_m, C_m	Modified graph and modified centrality
Z	Pairs of vertices to be covered
m_u	Number of uncovered pairs, $ Z $

Table 1: Frequently used symbols

version of the problem, where the goal is to optimize the coverage centrality of a target set of nodes by adding edges. When the target set has size one, optimization of different centralities was studied in [6, 12]. In [25], the authors solved a similar problem, which is maximizing the expected decrease in the sum of the shortest paths from a single source to the remaining nodes via edge addition.

Our Contributions. The main contributions of this paper can be summarized as follows:

- We study a novel network design problem, which is optimizing the coverage centrality of a group of nodes, and prove that it is NP-hard, even to approximate by a constant.
- We propose a simple greedy algorithm and an even faster sampling algorithm for group centrality maximization.
- We show the effectiveness of our algorithms on several real datasets and also prove that the proposed solutions are almost optimal for a constrained version of the problem.

2 PROBLEM DEFINITION

We assume $G(V, E)$ to be an undirected¹ graph with a set of vertices V and edges E . A shortest path between vertices s and t is a path with minimum distance (in hops) among all paths between s and t , with length $d(s, t)$. By convention, $d(s, s) = 0$, for all $s \in V$. Let P_{st} denote the set of vertices in the shortest paths (multiple ones might exist) between s and t where $s, t \notin P_{st}$. We define Z as the set of candidate pairs of vertices, $Z \subseteq V \setminus X \times V \setminus X$, which we want to cover. The *coverage centrality* of a vertex is defined as:

$$C(v) = |\{(s, t) \in Z | v \in P_{st}, s \neq v, t \neq v\}| \quad (1)$$

$C(v)$ gives the number of pairs of vertices with at least one shortest path going through (i.e. covered by) vertex v . The *coverage centrality* of a set of vertices $X \subseteq V$ is defined as:

$$C(X) = |\{(s, t) \in Z | v \in P_{st}, v \in X \wedge s, t \notin X\}| \quad (2)$$

A set X covers a pair (s, t) iff $X \cap P_{st} \neq \emptyset$, i.e., at least one vertex in X is part of a shortest path from s to t . Our goal is to maximize the coverage centrality of a given set X over a set of pairs Z by adding edges from a set of candidate edges Γ to G . For instance, in our online advertising example, X are pages owned by the car rental company, Z are pairs of pages related to travel and cars, and Γ are potential links that can be created to increase the traffic through the pages in X . Let G_m denote the modified graph after adding

¹We discuss how our methods can be generalized to directed networks in the Appendix.

edges $E_s \subseteq \Gamma$, $G_m = (V, E \cup E_s)$. We define the coverage centrality of X (over pairs in Z) in the modified graph G_m as $C_m(X)$.

PROBLEM 1. Coverage Centrality Optimization

(CCO): Given a network $G = (V, E)$, a set of vertices $X \subset V$, a candidate set of edges Γ , a set of vertex pairs Z and a budget k , find a set of edges $E_s \subseteq \Gamma$, such that $|E_s| \leq k$ and $C_m(X)$ is maximized.

For simplicity, in the rest of the paper, we assume $Z = V \setminus X \times V \setminus X$ unless stated otherwise. As a consequence:

$$C(X) = |\{(s, t) \in V \setminus X \times V \setminus X | v \in P_{st}, v \in X, s < t\}| \quad (3)$$

where $s < t$ implies ordered pairs of vertices. Fig. 2 shows a solution for the CCO problem with budget $k = 1$ for an example network where the target set $X = \{d, f\}$ and the candidate set $\Gamma = \{(d, a), (d, b), (f, b)\}$.

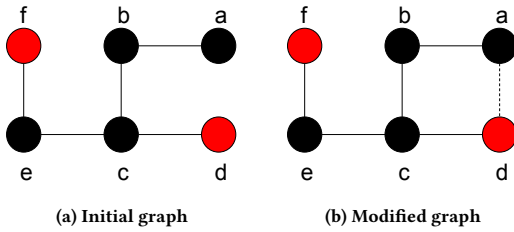


Figure 2: Example of Coverage Centrality Optimization problem. We want to optimize the centrality of $\{d, f\}$ with a budget of one edge from the candidates $\{(d, a), (d, b), (f, b)\}$. The coverage centrality of $\{d, f\}$ is 0 in the initial graph (a) and 3 in the modified graph (b). Node d belongs to the shortest paths between (a, e) , (a, c) and (a, f) in (b).

3 HARDNESS AND INAPPROXIMABILITY

This section provides complexity analysis of the CCO problem. We show that CCO is NP-hard, even to approximate by a constant. This result not only certifies that there are no efficient algorithms with constant quality guarantees for our general problem, but also offers a broader perspective on the hardness of network design problems.

THEOREM 1. *The CCO problem is NP-hard.*

PROOF. Consider an instance of the NP-complete Set Cover problem, defined by a collection of subsets S_1, S_2, \dots, S_m for a universal set of items $U = \{u_1, u_2, \dots, u_n\}$. The problem is to decide whether there exist k subsets whose union is U . To define a corresponding CCO instance, we construct an undirected graph with $m + 2n + 3$ nodes: there are nodes i and j corresponding to each set S_i and each element u_j respectively, and an undirected edge (i, j) whenever $u_j \in S_i$. Every S_i has an edge with S_j when $i \neq j$ and $i, j \in 1, 2, \dots, m$. Set $T = \{t_1, t_2, \dots, t_n\}$ is a copy of set U where u_i is connected to the corresponding t_i for all $i \in 1, 2, \dots, n$. Three more nodes (a, b and c) are added to the graph where a is in X . Node c is connected to t_i for all $i \in 1, 2, \dots, n$. Node b is attached to a and c . Figure 3 shows an example of this construction. The reduction clearly takes polynomial time. The candidate set Γ consists of the edges between a and set S . We prove that CCO of a given singleton set is NP-hard by maximizing the Coverage Centrality (CC) of the node a . Current CC of a is 0 by construction.

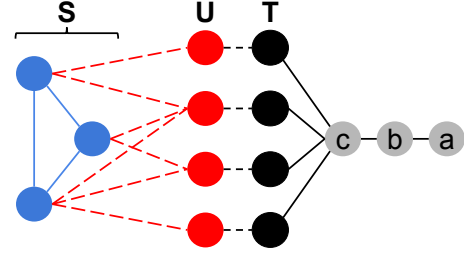


Figure 3: Example of reduction from Set Cover to Coverage Centrality Optimization, where $|U| = 4$ and $|S| = 3$. Target set $X = \{a\}$ and candidate edges Γ connect a to nodes in set S .

A set $S' \subset S$, with $|S'| \leq k$ is a set cover iff the CC of a becomes $n + m + k$ after adding the edges between a and every node in S' . Assume that S' is a set cover and edges are added between node a and every node in S' . Then the CC of a improves by $m + n + k$ as shortest paths between pairs (b, s) , $\forall s \in S$, (b, u) , $\forall u \in U$, and (c, p) , $\forall p \in S'$, will now pass through a . On the other hand, assume that the CC of node a is $m + n + k$ after adding edges between a and any set $S' \subset S$. It is easy to see that $m + k$ extra pairs will have their shortest paths covered by a . However, the only way to add another n pairs is by making S' a set cover. \square

Given that computing an optimal solution for CCO is infeasible in practice, a natural question is whether there is a polynomial-time approximation for the problem. The next theorem shows that CCO is also NP-hard to approximate by any constant. Interestingly, such a proof requires a more elaborate construction than the one used in the last theorem. The intuition behind this fact is that graph modification problems have a stronger combinatorial nature than traditional coverage problems. In particular, while maximum coverage can be approximated by a simple greedy algorithm [22], the same does not hold for our problem.

THEOREM 2. *CCO is NP-hard to approximate by a constant.*

PROOF. Let $SK(U, S, P, W, B)$ be an instance of the Set Union Knapsack Problem [9], where U is a set of items $\{u_1, u_2, \dots, u_n\}$, $S = S_1, S_2, \dots, S_m$ is a set of subsets ($S_i \subseteq U$), $p : S \rightarrow \mathbb{R}_+$ is a subset profit function, $w : U \rightarrow \mathbb{R}_+$ is an item weight function, and $B \in \mathbb{R}_+$ is the budget. For a subset $\mathcal{A} \subseteq S$, we define the weighted union of set \mathcal{A} as $W(\mathcal{A}) = \sum_{e \in \bigcup_{t \in \mathcal{A}} S_t} w_e$ and $P(\mathcal{A}) = \sum_{t \in \mathcal{A}} p_t$. The problem asks for subsets $S^* \subseteq S$ such that $W(S^*) \leq B$ and $P(S^*)$ is maximized. The problem is known to be NP-hard to approximate within a constant factor, even for unit weights and profits [1].

We reduce the unit profit and weight version of SK to the CCO problem. The graph G is constructed as follows. For each item $u_j \in U$, we create two vertices, a_j and b_j , in V . Moreover, for each subset S_i and item $u_j \in S_i$, we create pairs of vertices, $c_{i,j}$ and $d_{i,j}$. Three extra vertices, p_i , q_i , and x_i , for each subset S_i are also added to V . Regarding the edge set E , edges $(a_j, c_{i,j})$ and $(b_j, d_{i,j})$, for each corresponding set S_i and item u_j , are added to E . Assuming some arbitrary order $\langle u_1, u_2, \dots, u_{|S_i|} \rangle$ over the items in each set S_i , we also add $|S_i| - 1$ edges $(c_{i,r}, d_{i,r+1})$ for each S_i . Finally, we add edges (p_i, x_i) , $(x_i, c_{i,1})$, and $(q_i, d_{i,|S_i|})$, assuming

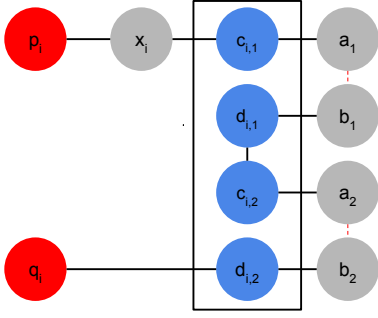


Figure 4: General structure of our reduction from Set Union Knapsack Problem to Coverage Centrality Optimization for a single set $S_i = \{u_1, u_2\}$.

the same ordering. Candidate edges in Γ are in the form (a_j, b_j) , for each item u_j . The set of pairs Z to be covered is composed of pairs p_j, q_j and the target vertices, for which the coverage centrality has to be increased, contains vertices x_i , for each subset S_i . The number of edges to be added k is set as the total budget B . Figure 4 illustrates the structure of our construction for a set $S_i = \{u_1, u_2\}$.

For any solution \mathcal{A} of an instance of unit SK there is a corresponding instance (G, X, Z, Γ, k) of the CCO problem with coverage of $C_m = P(\mathcal{A})$. This follows from the fact that each x_i will cover a path from p_i to q_i iff all the edges (a_j, b_j) corresponding to items in S_i are added. Moreover, as $G(V, E \cup \Gamma)$ is a tree, there is only one possible path between any pair (p_i, q_i) . This proves our claim. \square

Theorem 2 shows that there is no polynomial-time constant-factor approximation for CCO. Given such an impossibility result, the general approach is to resort to some heuristic that achieves good results in practice, as discussed in the next section.

4 ALGORITHMS

4.1 Greedy Algorithm (GES)

Algorithm 1 (GES) is a simple greedy strategy that selects the best edge to add in each of the k iterations, where k is the budget. Its most important steps are 2 and 7. In step 2, it computes all-pair-shortest-paths in time $O(n(m+n))$. Next, it chooses, among the candidate edges Γ , the one that maximizes the marginal coverage centrality gain of X (step 7), which takes $O(|\Gamma|n^2)$ time. After adding the best edge, the shortest path distances are updated. Then, the algorithm checks the pairwise distances in $O(n^2)$ time (step 9). The total running time of GES is $O(n(m+n) + k|\Gamma|n^2)$.

We illustrate the execution of GES on the graph from Figure 2a for a budget $k = 2$, a candidate set of edges $\Gamma = \{(d, a), (d, b), (f, b)\}$, and a target set $X = \{d, f\}$. Initially, adding (d, a) , (d, b) and (f, b) increases the centrality of X by 3, 0, and 2, respectively, and thus (d, a) is chosen. In the second iteration, (d, b) and (f, b) increase the centrality of X by 0 and 1, respectively, and (f, b) is chosen.

4.2 Sampling Algorithm (BUS)

The execution time of GES increases with $|\Gamma|$ and m . In particular, if $m = O(n^2)$ and $|\Gamma| = O(n)$, the complexity reaches $O(n^3)$, which is prohibitive for large graphs. To address this challenge, we propose a sampling algorithm that is nearly optimal, regarding each greedy

Algorithm 1: Greedy Edge Set (GES)

Require: Network $G = (V, E)$, target node set X , Candidate set of edges Γ , Budget k

Ensure: A subset E_s from Γ of k edges

```

1:  $E_s \leftarrow \emptyset$ 
2: Compute all pair shortest paths and store the distances
3: while  $|E_s| \leq k$  do
4:   for  $e \in \Gamma \setminus E_s$  do
5:      $\text{Count}(e) \leftarrow$  # newly covered pairs after adding  $e$ 
6:   end for
7:    $e^* \leftarrow \arg \max_{e \in \Gamma \setminus E_s} \{\text{Count}(e)\}$ 
8:    $E_s \leftarrow E_s \cup e^*$  and  $E \leftarrow E \cup e^*$ 
9:   Update the shortest path distances
10: end while
11: return  $E_s$ 
```

Algorithm 2: Best Edge via Uniform Sampling (BUS)

Require: Network $G = (V, E)$, target node set X , Candidate set of edges Γ , Budget k

Ensure: A subset γ from Γ of k edges

```

1: Choose  $q$  pairs of vertices in  $Q$  from  $M_u$ 
2:  $\gamma \leftarrow \emptyset$ 
3: while  $|\gamma| \leq k$  do
4:   for  $(s, t) \in Q$  do
5:     Compute and store s.p. distance  $d(s, v)$  (for all  $v \in V$ )
6:     Compute and store s.p. distance  $d(t, v)$  (for all  $v \in V$ )
7:   end for
8:   for  $e \in \Gamma \setminus \gamma$  do
9:      $\text{score}_e \leftarrow$  # newly covered pairs after adding  $e$ 
10:  end for
11:   $e^* \leftarrow \arg \max_{e \in \Gamma \setminus \gamma} \{\text{score}_e\}$ 
12:   $\gamma \leftarrow \gamma \cup e^*$  and  $E \leftarrow E \cup e^*$ 
13: end while
14: Return  $\gamma$ 
```

edge choice, with probabilistic guarantees (see Section 5.3). Instead of selecting edges based on all the uncovered pairs of vertices, our scheme does it based on a small number of sampled uncovered pairs. This strategy allows the selection of edges with probabilistic guarantees using a small number of samples, thus ensuring scalability to large graphs. We show that the error in estimating the coverage based on the samples is small.

Algorithm 2 (Best Edge via Uniform Sampling, or BUS) is a sampling strategy to select the best edge to be added in each of the k iterations based on the sampled uncovered node pairs. For each pair of samples, we compute the distances from each node in the pair to all others. These distances estimate the number of covered pairs after the addition of one edge. In Section 5.3, we provide a theoretical analysis of the approximation achieved by BUS.

The costliest steps of our algorithm are 4-7 and 8-10. Steps 4-7, where the algorithm performs shortest-path computations, take $O(q(n+m))$ time. Next, the algorithm estimates the additional number of shortest pairs covered by X after adding each of the edges based on the samples (steps 8-10) in $O(|\Gamma|q^2)$ time. Given such an estimate, the algorithm chooses the best edge to be added (step 11). The total running time of BUS is $O(kq(m+n) + k|\Gamma|q^2)$.

5 ANALYSIS

In the previous section, we described an efficient algorithm for CCO. Based on the inapproximability result from Theorem 2, such an algorithm cannot provide a constant-factor approximation for our general problem. Nevertheless, under some realistic assumptions, we show that the described algorithm provides a constant-factor approximation for a modified version of CCO. More specifically, our approximation guarantees are based on the addition of two extra constraints to the general problem described in Section 2.

5.1 Constrained Problem

The extra constraints, S^A and S^B , considered are the following: (1) S^A : We assume that edges are added from the target set X to the remaining nodes, i.e. edges in a given candidate set Γ have the form (a, b) where $a \in X$ and $b \in V \setminus X$ [6]; and (2) S^B : Each pair (s, t) can be covered by at most one single newly added edge [3, 21].

S^A is a reasonable assumption in many applications. For instance, in online advertising, adding links to a third-party page gives away control over the navigation, which is undesirable. S^B is motivated by the fact that, in real-life graphs, vertex centrality follows a skewed distribution (e.g. power-law), and thus most of the new pairs will have shortest paths through a single edge in Γ . In our experiments (see Table 4 in Section 6.1), we show that, in practice, solutions for the constrained and general problem with S^A are not far from each other. Both constraints have been considered by previous work [3, 6, 21]. Next, we show that CCO under constraints S^A and S^B , or RCCO (Restricted CCO), for short, is still NP-hard.

COROLLARY 3. *RCCO is NP-hard.*

PROOF. Follows directly from Theorem 1, as the construction applied in the proof respects both the constraints. \square

Notice that our inapproximability result (Theorem 2) applies a construction that violates the constraints S^A and S^B . This motivates us to explore the existence of approximate algorithms for RCCO.

5.2 Analysis: Greedy Algorithm

The next Theorem shows that the optimization function related to RCCO is monotone and submodular. As a consequence, a greedy approach, such as the algorithm described in Section 4.1, leads to a well-known constant factor approximation of $(1 - 1/e)$ [22].

THEOREM 4. *The objective function $f(E_s) = C_m(X)$ in RCCO is monotone and submodular.*

PROOF. Monotonicity: Follows from the definition of a shortest path. Adding an edge $(u, v) \in E_s$ cannot increase $d(s, t)$ for any (s, t) already covered by X . Since $u \in X$ for any $(u, v) \in E_s$, the coverage $C_m(X)$ is also non-decreasing.

Submodularity: We consider addition of two sets of edges, E_a and E_b where $E_a \subset E_b$, and show that $f(E_a \cup \{e\}) - f(E_a) \geq f(E_b \cup \{e\}) - f(E_b)$ for any edge $e \in \Gamma$ such that $e \notin E_a$ and $e \notin E_b$. Let $F(A)$ be the set of node pairs (s, t) which are covered by an edge $e \in A$ ($|F(E_s)| = C_m(X)$). Then $f(\cdot)$ is submodular if $F(E_b \cup \{e\}) \setminus F(E_b) \subseteq F(E_a \cup \{e\}) \setminus F(E_a)$. To prove this claim, we make use of S^B . Therefore, each pair $(s, t) \in F(E_b)$ is covered by only one edge in E_b . As $E_a \subset E_b$, adding e to E_a will cover some

of the pairs which are already covered by $E_b \setminus E_a$. Then, for any newly covered pair $(s, t) \in F(E_b \cup \{e\}) \setminus F(E_b)$, it must hold that $(s, t) \in F(E_a \cup \{e\}) \setminus F(E_a)$. \square

Based on Theorem 4, if OPT is the optimal solution for an instance of the RCCO problem, GES will return a set of edges E_s such that $f(E_s) \geq (1 - 1/e)OPT$. The existence of such an approximation algorithm shows that the constraints S^A and S^B make the CCO problem easier, compared to its general version. On the other hand, whether GES is a good algorithm for the modified CCO (RCCO) remains an open question. In order to show that our algorithm is almost optimal, in the sense that the best algorithm for this problem cannot achieve results far from those of GES, we also prove an inapproximability result for the constrained problem.

THEOREM 5. *RCCO cannot be approximated within a factor greater than $(1 - \frac{1}{4e})$.*

PROOF. We give an L -reduction [30] from the maximum coverage (MSC) problem with parameters x and y . Our reduction is such that following two equations are satisfied:

$$OPT(I_{RCCO}) \leq xOPT(I_{MSC}) \quad (4)$$

$$OPT(I_{MSC}) - s(T^M) \leq y(OPT(I_{RCCO}) - s(T^C)) \quad (5)$$

where I_{MSC} and I_{RCCO} are problem instances, and $OPT(Y)$ is the optimal value for instance Y . $s(T^M)$ and $s(T^C)$ denote any solution of the MSC and RCCO instances respectively. If the conditions hold and RCCO has an α approximation, then MSC has an $(1 - xy(1 - \alpha))$ approximation. However, MSC is NP-hard to approximate within a factor greater than $(1 - \frac{1}{e})$. It follows that $(1 - xy(1 - \alpha)) < (1 - \frac{1}{e})$, or, $\alpha < (1 - \frac{1}{xye})$ [6]. So, if the conditions are satisfied, RCCO is NP-hard to approximate within a factor greater than $(1 - \frac{1}{xye})$.

We use the same construction as in Theorem 1. For RCCO, the set Z contains pairs in the form (b, u) , $u \in U$.

Let the solution of I_{RCCO} be $s(T^C)$. It is easy to see that the centrality of node a will increase by $s(T^C)$ to cover the pairs in Z . Note that $s(T^C) = 2s(T^M)$ from the construction (as the graph is undirected, the covered pair is unordered). So, it follows that both the conditions are satisfied when $x = y = 2$. So, RCCO is NP-hard to approximate within a factor greater than $(1 - \frac{1}{4e})$. \square

While Theorem 5 does not certify that GES achieves the best approximation for the constrained CCO (RCCO) problem, it also does not imply that a better approximation algorithm exists. Instead, we use this Theorem as an evidence that the proposed algorithm is an almost optimal solution for the problem.

5.3 Analysis: Sampling Algorithm

In Section 4.2, we presented BUS, a fast sampling algorithm for the general CCO problem. Here, we study the quality of the approximation provided by BUS as a function of the number of sampled node pairs applied by the algorithm. The analysis will assume the constrained version of CCO (RCCO), but approximation guarantees regarding the general case will also be discussed.

Let us assume that X covers a set M_c of pairs of vertices. The set of remaining vertex pairs is M_u , $M_u = \{(s, t) | s \in V, t \in V, s \neq t, X \cap P_{st} = \emptyset\}$, $m_u = |M_u| = n(n-1) - |M_c|$. We sample, uniformly with replacement, a set of ordered vertex pairs Q ($|Q| = q$) from all

vertex pairs (M_u) not covered by X . Let $g^q(\cdot)$ denote the number of newly covered pairs by the candidate edges based on the samples Q . Moreover, for an edge set $\gamma \subset \Gamma$, let X_i be a random variable which denotes whether the i th sampled pair is covered by any edge in γ . In other words, $X_i = 1$ if the pair is covered and 0, otherwise. Each pair is chosen with probability $\frac{1}{m_u}$ uniformly at random.

LEMMA 5.1. *Given a size q sample of node pairs from M_u :*

$$E(g^q(\gamma)) = \frac{q}{m_u} f(\gamma)$$

From the sampling, we get $g^q(\gamma) = \sum_{i=1}^q X_i$. By the linearity and additive rule, $E(g^q(\gamma)) = \sum_{i=1}^q E(X_i) = q \cdot E(X_i)$. As the probability $P(X_i) = \frac{f(\gamma)}{m_u}$ and X_i s are i.i.d., $E(g^q(\gamma)) = \frac{q}{m_u} f(\gamma)$. We also define $f^q = \frac{m_u}{q} g^q$ as the estimated coverage based on samples.

LEMMA 5.2. *Given ϵ ($0 < \epsilon < 1$), a positive integer l , a budget k , and a sample of independent uncovered node pairs Q , $|Q| = q$, where $q(\epsilon) \geq \frac{3m_u(l+k)\log(|\Gamma|)}{\epsilon^2 \cdot OPT}$; then:*

$$Pr(|f^q(\gamma) - f(\gamma)| < \epsilon \cdot OPT) \geq 1 - 2|\Gamma|^{-l}$$

For all $\gamma \subset \Gamma$, $|\gamma| \leq k$, where OPT denotes the optimal coverage ($OPT = \max\{f(\gamma) | \gamma \subset \Gamma, |\gamma| \leq k\}$).

PROOF. Using Lemma 5.1:

$$\begin{aligned} Pr(|f^q(\gamma) - f(\gamma)| \geq \delta \cdot f(\gamma)) \\ Pr\left(\left|\frac{q}{m_u} f^q(\gamma) - \frac{q}{m_u} f(\gamma)\right| \geq \frac{q}{m_u} \cdot \delta \cdot f(\gamma)\right) \\ Pr\left(\left|g^q(\gamma) - \frac{q}{m_u} f(\gamma)\right| \geq \frac{q}{m_u} \cdot \delta f(\gamma)\right) \\ Pr(|g^q(\gamma) - E(g^q(\gamma))| \geq \delta E(g^q(\gamma))) \end{aligned}$$

As the samples are independent, applying Chernoff bound:

$$Pr\left(\left|g^q(\gamma) - \frac{q}{m_u} f(\gamma)\right| \geq \frac{q}{m_u} \delta f(\gamma)\right) \leq 2 \exp\left(-\frac{\delta^2}{3} \frac{q}{m_u} f(\gamma)\right)$$

Substituting $\delta = \frac{\epsilon \cdot OPT}{f(\gamma)}$ and q :

$$Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon \cdot OPT) \leq 2 \exp\left(-\frac{OPT}{f(\gamma)}(l+k)\log(\Gamma)\right)$$

Using the fact that $OPT \geq f(\gamma)$:

$$Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon \cdot OPT) \leq 2|\Gamma|^{-(l+k)}$$

Applying the union bound over all possible size- k subsets of $\gamma \subset \Gamma$ (there are $|\Gamma|^k$) we conclude the following:

$$Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon \cdot OPT) < 2|\Gamma|^{-l}, \forall \gamma \subset \Gamma$$

$$Pr(|f^q(\gamma) - f(\gamma)| < \epsilon \cdot OPT) \geq 1 - 2|\Gamma|^{-l}, \forall \gamma \subset \Gamma \quad \square$$

Now, we prove our main theorem which shows an approximation bound of $(1 - \frac{1}{e} - \epsilon)$ by Algorithm 2 whenever the number of samples is at least $q(\epsilon/2) = \frac{12m_u(l+k)\log(|\Gamma|)}{\epsilon^2 \cdot OPT}$ (l and ϵ are as in Lemma 5.2).

THEOREM 6. *Algorithm 2 ensures $f(\gamma) \geq (1 - \frac{1}{e} - \epsilon)OPT$ with high probability $(1 - \frac{2}{|\Gamma|^l})$ using at least $q(\epsilon/2)$ samples.*

PROOF. $f(\cdot)$ is monotonic and submodular (Thm. 4) and one can prove the same for $f^q(\cdot)$. Given the following:

- (1) Lemma 5.2: The number of samples is at least $q(\epsilon/2)$. So, with probability $1 - \frac{2}{|\Gamma|^l}$, $f(\gamma) \geq f^q(\gamma) - \frac{\epsilon}{2}OPT$;
- (2) $f^q(\gamma) \geq (1 - \frac{1}{e})f^q(\gamma^*)$, $\gamma^* = \arg \max_{\gamma' \subset \Gamma, |\gamma'| \leq k} f^q(\gamma')$ (submodularity property of $f^q(\cdot)$);
- (3) $f^q(\gamma^*) \geq f^q(\bar{\gamma})$, $\bar{\gamma} = \arg \max_{\gamma' \subset \Gamma, |\gamma'| \leq k} f(\gamma')$ (Note that, $OPT = f(\bar{\gamma})$)

We can prove with probability $1 - \frac{2}{|\Gamma|^l}$ that:

$$\begin{aligned} f(\gamma) &\geq f^q(\gamma) - \frac{\epsilon}{2}OPT \\ &\geq \left(1 - \frac{1}{e}\right)f^q(\gamma^*) - \frac{\epsilon}{2}OPT \\ &\geq \left(1 - \frac{1}{e}\right)f^q(\bar{\gamma}) - \frac{\epsilon}{2}OPT \\ &\geq \left(1 - \frac{1}{e}\right)\left(f(\bar{\gamma}) - \frac{\epsilon}{2}OPT\right) - \frac{\epsilon}{2}OPT \\ &> \left(1 - \frac{1}{e} - \epsilon\right)OPT \quad \square \end{aligned}$$

While we are able to achieve a good probabilistic approximation with respect to the optimal value OPT , deciding the number of samples is not straightforward. In practice, we do not know the value of OPT beforehand, which affects the number of samples needed. However, notice that OPT is bounded by the number of uncovered pairs m_u . Moreover, the number of samples $q(\epsilon/2)$ depends on the ratio $\frac{m_u}{OPT}$. Thus, increasing this ratio while keeping the quality constant requires more samples. Also, if OPT (which depends on X) is close to the number of uncovered pairs m_u , we need fewer samples to achieve the mentioned quality. In the experiments, we assume this ratio to be constant. Next, we propose another approximation scheme where we can reduce the number of samples by avoiding the term OPT in the sample size while waiving the assumption involving constants.

Let M_u and m_u be the set and number of uncovered pairs by X respectively in the initial graph. Let us assume,

$$\bar{q}(\epsilon) \geq \frac{3(l+k)\log(|\Gamma|)}{\epsilon^2}$$

COROLLARY 7. *Given ϵ ($0 < \epsilon < 1$), a positive integer l , a budget k , and a sample of independent uncovered node pairs Q , $|Q| = \bar{q}(\epsilon)$, then:*

$$Pr(|f^q(\gamma) - f(\gamma)| < \epsilon \cdot m_u) \geq 1 - 2|\Gamma|^{-l}, \forall \gamma \subset \Gamma, |\gamma| \leq k$$

The proof is given in the Appendix. Next, we provide an approximation bound by our sampling scheme for at least $\bar{q}(\epsilon/2) = \frac{12(l+k)\log(|\Gamma|)}{\epsilon^2}$ samples.

COROLLARY 8. *Algorithm 2 ensures $f(\gamma) \geq (1 - \frac{1}{e})OPT - \epsilon \cdot m_u$ with high probability $(1 - \frac{2}{|\Gamma|^l})$ for $\bar{q}(\epsilon/2)$ samples.*

This proof is also in the Appendix. Table 2 summarizes the number of samples and corresponding bounds for Algorithm 2. Theorem 6 ensures higher quality with higher number of samples than provided by Corollary 8. On the other hand, Corollary 8 does not assume anything about the ratio $\frac{m_u}{OPT}$. The results reflect a trade-off between the number of samples and the accuracy.

Thm.	#Samples	Approximations
Thm. 6	$O(\frac{m_u k \log(\Gamma)}{\epsilon^2 OPT})$	$f(\gamma) > (1 - \frac{1}{e} - \epsilon)OPT$
Cor. 8	$O(\frac{k \log(\Gamma)}{\epsilon^2})$	$f(\gamma) > (1 - \frac{1}{e})OPT - \epsilon \cdot m_u$

Table 2: #Samples and approximations with prob. $(1 - \frac{2}{|\Gamma|})$.

Dataset Name	V	E
ca-GrQc (CG)	5K	14K
email-Enron (EE)	36K	183K
loc-Brightkite (LB)	58K	214K
loc-Gowalla (LG)	196K	950K
web-Stanford (WS)	280K	2.3M
DBLP (DB)	1.1M	5M

Table 3: Dataset description and statistics.

Theorem 6 and Corollary 8 assume a greedy approach achieves a constant-factor approximation of $(1 - 1/e)$, which holds only for the RCCO problem (see Sections 5.1 and 5.2). However, the same approximation cannot be achieved in polynomial-time for the general CCO problem presented in Section 2, as shown in Theorem 2. As a consequence, in the case of the general problem, the guarantees discussed in this Section apply only for each iteration of our sampling algorithm, but not for the final results. In other words, BUS provides theoretical quality guarantees that each edge selected in an iteration of the algorithm achieves a coverage within bounded distance from the optimal edge. Nonetheless, experimental results show, in practice, BUS is also effective in the general setting.

6 EXPERIMENTAL RESULTS

Experimental Setup and Data: We evaluate the quality and scalability of our algorithms on real-world networks. All experiments were conducted on a 3.30GHz Intel Core i7 machine with 30 GB RAM. Algorithms were implemented in Java and all datasets applied are available online². Table 3 shows dataset statistics. The graphs are undirected and we consider the largest connected component for our experiments. Results reported are averages of 10 repetitions.

We set the candidate of edges Γ as those edges from X to the remaining vertices that are absent in the initial graph (i.e. $\Gamma = \{(u, v) | u \in X \wedge v \in V \setminus X \wedge (u, v) \notin E\}$). The given set of target nodes (set X) is randomly selected from the set of all nodes.

Baselines: We consider three baselines in our experiments: 1) **High-ACC:** Applies *maximum adaptive centrality coverage* [19, 31] and adds edges between target nodes X and the top- k centrality set; 2) **High-Degree:** Selects edges between the target nodes X and the top k high degree nodes; 3) **Random:** Randomly chooses k edges from Γ which are not present in the graph. We also compare our sampling algorithm (BUS) against our Greedy solution (GES) and show that BUS is more efficient while producing similar results.

6.1 GES: RCCO vs CCO

First, we compare coverage centrality optimization (CCO) and its restricted version (RCCO) empirically by applying our greedy algorithm (GES) to two small datasets: a co-authorship (NetScience) and a synthetic (Barabasi) network. The target set size $|X|$ is set to 5 in these experiments. Table 4 shows the ratio between results

Data	Ratio		
	$k = 5$	$k = 10$	$k = 15$
Co-authorship	1.02	1.14	1.17
Synthetic	1.0	1.0	1.0

Table 4: The ratio between the solutions produced by GES for the general (CCO) and constrained (RCCO) settings.

for CCO and RCCO varying the budget k . As the ratios are close to 1, one can conclude that CCO and RCCO are similar in practice, which motivates the study of RCCO.

6.2 BUS vs. GES

Here, we apply only the smallest dataset (CG) as the GES algorithm is not scalable. Because all possible pairs are considered, we have to compute all-pair-shortest-paths to evaluate how many pairs are covered by each algorithm. For BUS, we set the error $\epsilon = 0.3$. First, we evaluate the effect of sampling on quality, which we theoretically analyzed in Theorem 6 and Corollary 8.

Fig. 5a shows the number of new pairs covered by the algorithms. Table 5 shows the running times and the quality of BUS relative to the baselines —i.e. how many times more pairs are covered by BUS compared to a given baseline. BUS and GES produce results at least 2 times better than the baselines. Moreover, BUS achieves results comparable to GES while being 2-3 orders of magnitude faster.

6.3 Results for Large Graphs

In this section, we compare our sampling-based algorithm against the baseline methods using large graphs (EE, LB, LG, WS and DB). Due to the high cost of computing all-pairs shortest-paths, we estimate the coverage centrality based on 10000 randomly selected pairs. For High-ACC, we also use sampling for adaptive coverage centrality computation [19, 31] and the same number of samples is used by High-ACC and BUS. The budget and target set size are fixed at 20 edges and 5 nodes, respectively.

Table 6 shows the results, where the quality is relative to BUS results. BUS takes a few minutes (8, 15, 17, 45, 85 minutes for EE, LB, WS, LG and DB respectively) to run and significantly outperforms the baselines. This is due to the fact that existing approaches do not take into account the dependencies between the edges selected in the coverage centrality. BUS selects the edges sequentially, considering the effect of edges selected in previous steps.

6.4 Parameter Sensitivity

The main parameters of BUS are the budget and the number of samples —both affect the error ϵ , as discussed in Thm. 6 and Cor. 8. We study the impact of these two parameters on performance. Again, we estimate coverage using 10000 randomly selected pairs of nodes. First, we fix the budget and vary the number of samples.

Figure 5b shows the results on EE data for budget 20 and target set size 5. With 600 samples, BUS produces results at least 2 times better than the baselines. Next, we fix the number of samples and vary the budget. Figure 5c shows the results on EE data with 1000 samples and 5 target nodes. BUS produces results at least 2.5 times better than the baselines. As expected, its running time increases with the number of samples and budget. BUS takes only 30 seconds to run with budget of 30 and 1000 samples. We find that the running

²Datasets collected from (1) <https://snap.stanford.edu/data/index.html>, (2) <http://dblp.uni-trier.de>, (3) <http://www-personal.umich.edu/~mejn/netdata/>

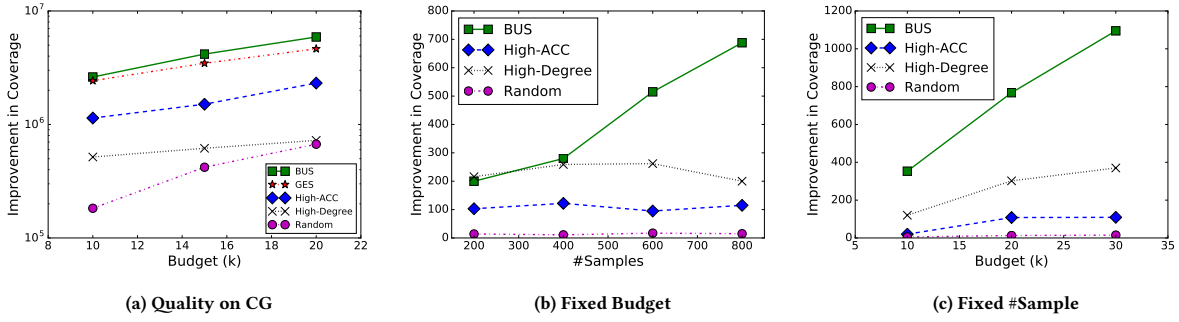


Figure 5: (a) BUS vs. Greedy: Improved coverage centrality produced by different algorithms on the CG dataset. (b-c) Comparison with baselines on the EE dataset varying (b) the number of samples and (c) the budget.

Budget	Coverage of BUS (relative to baselines)				Time [sec.]		# Samples	
	GES	High-ACC	High-Degree	Random	GES	High-ACC	BUS	BUS
$k = 10$	1.08	2.46	5.41	14.45	> 7200	157.1	5.1	2560
$k = 15$	1.21	2.92	7.29	9.98	> 7200	156.9	10.1	3840
$k = 20$	1.29	2.78	9.96	9.59	> 7200	157.2	18.2	5120

Table 5: Comparison between our sampling algorithm (BUS) and the baselines, including our Greedy (GES) approach, using the CG dataset and varying the budget k . We evaluate the coverage of BUS relative to the baselines —i.e. how many times more new pairs are covered by BUS compared to the baseline.

Data	Coverage of BUS (relative to baselines)			# Samples
	High-Acc	High-Degree	Random	BUS
EE	4.88	2.74	51	6462
LB	3.3	2.3	33.8	6796
LG	3.3	4.2	62	4255
WS	1.89	1.95	4.8	2000
DB	2.5	1.6	5	875

Table 6: BUS vs. baselines for large datasets.

k	Influence			Distance			Closeness		
	EE	LB	LG	EE	LB	LG	EE	LB	LG
25	57.7	12.2	10.7	2.7	1.2	2.2	2.0	2.0	1.0
50	96.8	17.5	92.7	3.8	3.5	3.3	4.9	3.9	4.0
75	134.3	29.1	45.9	5.2	2.1	2.3	5.9	2.3	1.9

Table 7: Improvement of other metrics after adding the edges found by BUS: the numbers are improvement in percentage with respect to the value for the initial graph.

time grows linearly with the budget for a fixed number of samples. These results validate the running time analysis from Section 4.2.

6.5 Impact on other Metrics

While this paper is focused on optimizing Coverage Centrality, it is interesting to analyze how our methods affect other relevant metrics. Here, we look at the following ones: 1) influence, 2) average shortest-path distance, and 3) closeness centrality. The idea is to assess how BUS improves the influence of the target nodes, decreases the distances from the target to the remaining nodes, and increases the closeness centrality of these nodes as new edges are added to the graph. For influence analysis, we consider the popular independent cascade model [13] assuming edge probabilities as 0.1. In all the experiments, we fix the number of sampled pairs at 1000 and choose 10 nodes, uniformly at random, as the target set X . The metrics are computed before and after the addition of edges and presented as the relative improvement in percentage. Notice that

because target nodes are chosen at random, increasing the budget does not necessarily lead to an increase in the metrics considered.

Results are presented in Table 7. There is a significant improvement of the three metrics as the budget (k) increases. For influence, the number of seed nodes is small, and thus the relative improvement for increasing k is large. The improvement of the other metrics is also significant. For instance, in EE, the decrease in distance is nearly 5%, which is approximately 72K, for a budget of 75.

7 CONCLUSIONS

In this paper, we studied a novel network design problem, the group centrality optimization. This problem has applications in a variety of domains including social, collaboration, and communication networks. We proved that the problem is NP-hard to approximate within a constant factor. We proposed a simple greedy algorithm and a faster randomized algorithm based on sampling. We further studied the problem under restricted yet realistic settings. In the restricted setting, the problem is APX-hard but our proposed greedy approach achieves a nearly optimal approximation factor. The randomized technique also obtains a probabilistic approximation guarantee. We evaluated our approaches on several real-world graphs showing that it outperforms the best baseline solution in terms of quality —coverage centrality achieved— by up to 5 times.

As future work, we will investigate the dynamic version of the problem [11, 17, 28], where coverage centrality has to be maintained under temporal, and possibly adversarial, edge updates. This problem has interesting connections with existing work on *Game Theory* [5]. Moreover, we will study other design problems that optimize social influence and consensus in networks [3, 14].

REFERENCES

- [1] Ashwin Arulselvan. 2014. A note on the set union knapsack problem. *Discrete Applied Mathematics* 169 (2014), 214–218.

- [2] Ulrik Brandes. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology* (2001), 163–177.
- [3] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *WWW*. 529–538.
- [4] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*. ACM, 1029–1038.
- [5] E. N. Ciftcioglu, S. Pal, K. S. Chan, D. H. Cansever, A. Swami, A. Singh, and P. Basu. 2016. Topology design under adversarial dynamics. In *WiOpt*. IEEE, 1–8.
- [6] Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2015. Greedily Improving Our Own Centrality in A Network. In *SEA*. Springer International Publishing, 43–55.
- [7] E. D. Demaine and M. Zadimoghaddam. 2010. Minimizing the diameter of a network using shortcut edges. *SWAT, ser. Lecture Notes in Computer Science*, H. Kaplan, Ed. (2010), 420–431.
- [8] Bistra Dilkina, Katherine J. Lai, and Carla P. Gomes. 2011. Upgrading shortest paths in networks. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, 76–91.
- [9] Olivier Goldschmidt, David Nehme, and Gang Yu. 1994. Note: On the set-union knapsack problem. *Naval Research Logistics (NRL)* 41, 6 (1994), 833–842.
- [10] Anupam Gupta and Jochen Könnemann. 2011. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science* (2011), 3–20.
- [11] Takanori Hayashi, Takuya Akiba, and Yuichi Yoshida. 2015. Fully dynamic betweenness centrality maintenance on massive networks. *Proceedings of the VLDB Endowment* 9, 2 (2015), 48–59.
- [12] Vatche Ishakian, Dóra Erdős, Evimaria Terzi, and Azer Bestavros. 2012. A Framework for the Evaluation and Management of Network Centrality. In *SDM*. SIAM, 427–438.
- [13] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*. 137–146.
- [14] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable Diffusion-aware Optimization of Network Topology. In *KDD*. ACM, 1226–1235.
- [15] Masahiro Kimura and Kazumi Saito. 2006. Tractable models for information diffusion in social networks. In *PKDD*. 259–271.
- [16] Sven O. Krumke, Madhav V. Marathe, Hartmut Noltemeier, R. Ravi, and S.S. Ravi. 1998. Approximation Algorithms for Certain Network Improvement Problems. *Journal of Combinatorial Optimization* 2 (1998), 257–288.
- [17] Kristina Lerman, Rumi Ghosh, and Jeon Hyung Kang. 2010. Centrality metric for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM, 70–77.
- [18] Yimin Lin and Kyriakos Mouratidis. 2015. Best upgrade plans for single and multiple source-destination pairs. *GeoInformatica* 19, 2 (2015), 365–404.
- [19] Ahmad Mahmoody, E Charalampous, and Eli Upfal. 2016. Scalable Betweenness Centrality Maximization via Sampling. In *KDD*. ACM.
- [20] Sourav Medya, Petko Bogdanov, and Ambuj Singh. 2016. Towards Scalable Network Delay Minimization. In *ICDM*. IEEE, 1083–1088.
- [21] A. Meyerson and B Tagiku. 2009. Minimizing average shortest path distances via shortcut edge addition. In *APPROX-RANDOM, I. Dinur, K. Janson, J. Naor and J. D. P. Rolim Eds, Vol. 5687*. Springer, 272–285.
- [22] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. Best Algorithms for Approximating the Maximum of a Submodular Set Function. *Math. Oper. Res.* (1978), 177–188.
- [23] D. Paik and S. Sahni. 1995. Network upgrading problems. *Networks* (1995), 45–58.
- [24] N Parotisidis, Evangelia Pitoura, and Panayiotis Tsaparas. 2015. Selecting shortcuts for a smaller world. In *SDM*. SIAM, 28–36.
- [25] Nikos Parotisidis, Evangelia Pitoura, and Panayiotis Tsaparas. 2016. Centrality-Aware Link Recommendations. In *WSDM*. ACM, 503–512.
- [26] Senni Perumal, Prithwish Basu, and Ziyu Guan. 2013. Minimizing Eccentricity in Composite Networks via Constrained Edge Additions. In *MILCOM*. 1894–1899.
- [27] Matteo Riondato and Evgenios M Kornaropoulos. 2014. Fast approximation of betweenness centrality through sampling. In *WSDM*. ACM, 413–422.
- [28] Taro Takaguchi, Yosuke Yano, and Yuichi Yoshida. 2016. Coverage centralities for temporal networks. *The European Physical Journal B* 89, 2 (2016), 1–11.
- [29] Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2012. Gelling, and Melting, Large Graphs by Edge Manipulation. In *CIKM*. ACM, 245–254.
- [30] David P Williamson and David B Shmoys. 2011. *The design of approximation algorithms*. Cambridge university press.
- [31] Yuichi Yoshida. 2014. Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches. In *KDD*. ACM, 1416–1425.
- [32] Qing K Zhu. 2004. *Power distribution network design for VLSI*. John Wiley & Sons.

APPENDIX

Proof of Corollary 7

Using Lemmas 5.1 and 5.2:

$$\begin{aligned} \Pr(|f^q(\gamma) - f(\gamma)| \geq \delta \cdot f(\gamma)) \\ \Pr(|g^q(\gamma) - E(g^q(\gamma))| \geq \delta E(g^q(\gamma))) \end{aligned}$$

The rest of the proof follows that for Lemma 5.2 but replacing *OPT* by m_u . As the samples are independent, we can apply the Chernoff bound:

$$\Pr\left(|g^q(\gamma) - \frac{\bar{q}}{m_u} f(\gamma)| \geq \frac{\bar{q}}{m_u} \delta f(\gamma)\right) \leq 2 \exp\left(-\frac{\delta^2}{3} \frac{\bar{q}}{m_u} f(\gamma)\right)$$

Now, substituting $\delta = \frac{\epsilon \cdot m_u}{f(\gamma)}$ and \bar{q} :

$$\Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon \cdot m_u) \leq 2 \exp\left(-\frac{m_u}{f(\gamma)}(l + k) \log(\Gamma)\right)$$

Using the fact that $m_u \geq f(\gamma)$:

$$\Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon m_u) \leq 2|\Gamma|^{-(l+k)}$$

Now, we apply the union bound over all possible size- k subsets of $\gamma \subset \Gamma$ (there are $|\Gamma|^k$) to get the following:

$$\begin{aligned} \Pr(|f^q(\gamma) - f(\gamma)| \geq \epsilon \cdot m_u) &< 2|\Gamma|^{-l}, \forall \gamma \subset \Gamma \\ \Pr(|f^q(\gamma) - f(\gamma)| < \epsilon \cdot m_u) &\geq 1 - 2|\Gamma|^{-l}, \forall \gamma \subset \Gamma \end{aligned}$$

Proof of Corollary 8

By the same arguments as in the proof of Theorem 6, with probability $1 - \frac{2}{|\Gamma|^l}$:

$$\begin{aligned} f(\gamma) &\geq f^q(\gamma) - \frac{\epsilon}{2} m_u \\ &\geq \left(1 - \frac{1}{e}\right) f^q(\gamma^*) - \frac{\epsilon}{2} m_u \\ &\geq \left(1 - \frac{1}{e}\right) f^q(\bar{\gamma}) - \frac{\epsilon}{2} m_u \\ &\geq \left(1 - \frac{1}{e}\right) \left(f(\bar{\gamma}) - \frac{\epsilon}{2} m_u\right) - \frac{\epsilon}{2} m_u \\ &= \left(1 - \frac{1}{e}\right) OPT - \left(\epsilon - \frac{\epsilon}{2e}\right) m_u \\ &> \left(1 - \frac{1}{e}\right) OPT - \epsilon m_u \end{aligned}$$

Towards More General Settings

We start by extending our approaches to solve the Coverage Centrality optimization problem on *directed* graphs (e.g. our motivating example in Figure 1). In this setting, edges are added from or towards the target nodes X —i.e. directed edges in Γ are of the form (u, v) or (v, u) where $u \in V \setminus X, v \in X$.

PROBLEM 2. Coverage Centrality Optimization in Directed Graphs (CCO-D): Given a directed network $G = (V, E)$, a target set of nodes $X \subset V$, a candidate set of edges Γ , and a budget k , find a set of edges $E_s \subset \Gamma$, such that $|E_s| \leq k$ and $C_m(X)$ is maximized.

We assume the same constraint S^B (as in the undirected graphs) on CCO-D. The next theorem shows that it is NP-hard to approximate within a factor greater than $(1 - \frac{1}{e})$ even under this constraint.

THEOREM 9. CCO-D under S^B cannot be approximated within a factor greater than $(1 - \frac{1}{e})$.

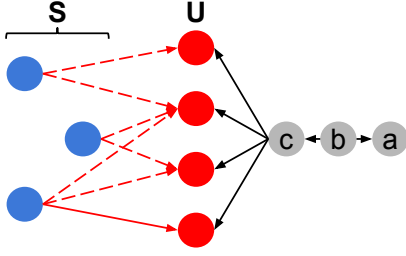


Figure 6: Example of reduction from Maximum Set Coverage, where $|U| = 4$ and $|S| = 3$. Target set $X = \{a\}$ and candidate edges Γ connect a to nodes in set S .

PROOF. We give a L -reduction [30] from the maximum coverage (MSC) problem with parameters x and y . Our reduction is such that following two equations are satisfied:

$$OPT(I_{CD}) \leq xOPT(I_{MSC}) \quad (6)$$

$$OPT(I_{MSC}) - s(T^M) \leq y(OPT(I_{CD}) - s(T^C)) \quad (7)$$

where I_{MSC} and I_{CD} are the two problem instances, OPT denotes the optimal values of the optimization problem instances. $s(T^M)$ and $s(T^C)$ denote any solution of the MSC and CCO-D instances respectively. If the conditions hold and CCO-D has an α approximation, then MSC has an $(1 - xy(1 - \alpha))$ approximation algorithm. However, MSC is NP-hard to approximate within a factor greater than $(1 - \frac{1}{e})$. It follows that $(1 - xy(1 - \alpha)) < (1 - \frac{1}{e})$, or, $\alpha < (1 - \frac{1}{xye})$ [6]. So, if the above two conditions are satisfied then CCO-D is NP-hard to approximate within a factor greater than $(1 - \frac{1}{xye})$.

Consider an instance of the Maximum Coverage (MSC) problem, defined by a collection of subsets S_1, S_2, \dots, S_m for a universal set of items $U = \{u_1, u_2, \dots, u_n\}$. To define a corresponding CCO-D instance, we construct an directed graph with $m + n + 3$ nodes: there are nodes i and j corresponding to each set S_i and each element u_j respectively, and an directed edge (i, j) whenever $u_j \in S_i$. Three more nodes (a, b and c) are added to the graph where a is in X . Node c is connected to u_i by (c, u_i) for all $i = 1, 2, \dots, n$. Node b is attached to a by (b, a) and c by (b, c) . Figure 6 shows an example of this construction. The reduction clearly takes polynomial time. The candidate set Γ consists of the edges between a and set S . Here for CCO-D, the set to cover, Z is the set of pairs in the form (b, u) where $u \in U$.

Let the solution of I_{CD} be $s(T^C)$. It is easy to see that the centrality of node a will increase by $s(T^C)$ to cover the pairs in Q . Note that $s(T^C) = s(T^M)$ from the construction. So, it follows that both the conditions are satisfied when $x = y = 1$. So, CCO-D is NP-hard to approximate within a factor greater than $(1 - \frac{1}{e})$. \square

The next theorem shows that the objective function associated with Problem 2 under S^B is also monotone and submodular, as was the case for the undirected setting.

THEOREM 10. *Given X , the objective function, $f(E_s) = C_m(X)$ in CCO-D is monotone and submodular.*

The proof for Theorem 10 is similar to that for Theorem 4. Based on this Theorem, our algorithm (BUS) can be applied to solve CCO-D with similar guarantees. In other words, our approach is agnostic

to the direction of edges. Interestingly, Theorem 9 and Theorem 10 certify that GES achieves the best approximation for the constrained CCO-D problem.

We also briefly describe the CCO problem under different settings. In particular, we focus on possible restrictions on the set of candidate edges Γ . For *undirected graphs*, S^1 : Γ is a subset of the set of all absent edges, S^2 : Γ consists of absent edges of the form (u, v) where either u or v belongs to the target set X , and S^B : a pair is covered using at most one newly added edge. For *directed graphs*, S^4 : Γ is a subset of the set of all absent edges with arbitrary direction and S^5 Γ consists of absent edges of the form (u, v) where either u or v belongs to X , with any direction. The hardness of these problems can be assessed with variations of the reasoning applied in Theorem 1. Table 8 summarizes the different problem settings. We have already proven that the objective function is submodular for S^2 and S^B (undirected) and for S^5 and S^B (directed). Additionally, we prove that the objective functions for S^2 and S^5 are not submodular.

	Undirected			Directed		
Settings	S^1	S^2	S^2, S^B	S^4	S^5	S^5, S^B
Submodularity	No	No	Yes	No	No	Yes

Table 8: Summary of CCO problem under different settings.

Proof of Non-submodularity under S^2 and S^5 :

These are two counter examples (Fig. 7).

For S^2 : consider the following, $T = \{(x, a)\}$, $S = \{a\}$, $e = (x, b)$ and the target node is x . Clearly $S \subset T$ and $f(T) = f(S) = 0$. But $f(T \cup \{e\}) = 1$ as x is covering the pair (a, b) , whereas $f(S \cup \{e\}) = 0$. So, $f(T \cup \{e\}) - f(T) > f(S \cup \{e\}) - f(S)$, and, f is not submodular.

For S^5 : the proof is similar to S^2 . Let $T = \{(a, x)\}$, $S = \{a\}$, $e = (x, b)$ and the target node be x . Clearly $S \subset T$ and $f(T) = f(S) = 0$. But $f(T \cup \{e\}) = 1$ as x is covering the pair (a, b) , whereas $f(S \cup \{e\}) = 0$. So, $f(T \cup \{e\}) - f(T) > f(S \cup \{e\}) - f(S)$, and, f is not submodular.

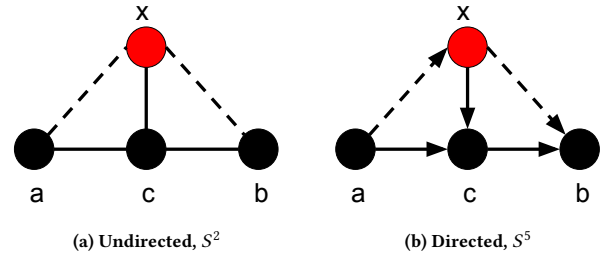


Figure 7: Constructions to prove non-submodularity under (a) S^2 and (b) S^5 .